D 8

```
RRRRRRRRRRRR     PPPPPPPPPPPP        GGGGGGGGGGGG  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
RRRRRRRRRRRR     PPPPPPPPPPPP        GGGGGGGGGGGG  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
RRRRRRRRRRRR     PPPPPPPPPPPP        GGGGGGGGGGGG  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
RRR         RRR  PPP         PPP  GGG              RRR         RRR        TTT        LLL
RRR         RRR  PPP         PPP  GGG              RRR         RRR        TTT        LLL
RRR         RRR  PPP         PPP  GGG              RRR         RRR        TTT        LLL
RRR         RRR  PPP         PPP  GGG              RRR         RRR        TTT        LLL
RRR         RRR  PPP         PPP  GGG              RRR         RRR        TTT        LLL
RRR         RRR  PPP         PPP  GGG              RRR         RRR        TTT        LLL
RRRRRRRRRRRR     PPPPPPPPPPPP     GGG              RRRRRRRRRRRR           TTT        LLL
RRRRRRRRRRRR     PPPPPPPPPPPP     GGG              RRRRRRRRRRRR           TTT        LLL
RRRRRRRRRRRR     PPPPPPPPPPPP     GGG              RRRRRRRRRRRR           TTT        LLL
RRR   RRR        PPP              GGG   GGGGGGGGG  RRR   RRR              TTT        LLL
RRR   RRR        PPP              GGG   GGGGGGGGG  RRR   RRR              TTT        LLL
RRR   RRR        PPP              GGG   GGGGGGGGG  RRR   RRR              TTT        LLL
RRR      RRR     PPP              GGG         GGG  RRR      RRR           TTT        LLL
RRR      RRR     PPP              GGG         GGG  RRR      RRR           TTT        LLL
RRR      RRR     PPP              GGG         GGG  RRR      RRR           TTT        LLL
RRR         RRR  PPP                 GGGGGGGGG     RRR         RRR        TTT        LLLLLLLLLLLLLLL
RRR         RRR  PPP                 GGGGGGGGG     RRR         RRR        TTT        LLLLLLLLLLLLLLL
RRR         RRR  PPP                 GGGGGGGGG     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

```
_$

Va
--
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
```

```
**FILE**ID**RPGSQRT                                   E  4



RRRRRRRR    PPPPPPPP      GGGGGGGG    SSSSSSSS    QQQQQQ    RRRRRRRR    TTTTTTTTTT
RRRRRRRR    PPPPPPPP      GGGGGGGG    SSSSSSSS    QQQQQQ    RRRRRRRR    TTTTTTTTTT
RR      RR  PP      PP  GG          SS          QQ      QQ  RR      RR      TT
RR      RR  PP      PP  GG          SS          QQ      QQ  RR      RR      TT
RR      RR  PP      PP  GG          SS          QQ      QQ  RR      RR      TT
RR      RR  PP      PP  GG          SS          QQ      QQ  RR      RR      TT
RRRRRRRR    PPPPPPPP    GG              SSSSSS  QQ      QQ  RRRRRRRR        TT
RRRRRRRR    PPPPPPPP    GG              SSSSSS  QQ      QQ  RRRRRRRR        TT
RR  RR      PP          GG  GGGGGG          SS  QQ  QQ  QQ  RR  RR          TT
RR  RR      PP          GG  GGGGGG          SS  QQ  QQ  QQ  RR  RR          TT
RR    RR    PP          GG      GG          SS  QQ    QQ    RR    RR        TT          ....
RR    RR    PP          GG      GG          SS  QQ    QQ    RR    RR        TT          ....
RR      RR  PP            GGGGGG    SSSSSSSS      QQQQ  QQ  RR      RR      TT          ....
RR      RR  PP            GGGGGG    SSSSSSSS      QQQQ  QQ  RR      RR      TT          ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

RPG
1-0

```
0 MODULE RPG$SQRT ( %TITLE 'Get square root'
0                   IDENT = '1-002'          ! file: RPGSQRT.B32 EDIT:DG1002
0                   ) =
1 BEGIN
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1
1 !++
1 ! FACILITY:     RPGII SUPPORT
1 !
1 ! ABSTRACT
1 !
1 !       This routine supports the RPG SQRT opcode.
1 !
1 !
1 ! ENVIRONMENT:  Vax-11 User Mode
1 !
1 ! AUTHOR: Debess Grabazs, CREATION DATE: 8-Feb-1983
1 !
1 ! MODIFIED BY:
1 !
1 ! 1-001 - Original. DG 8-Feb-1983
1 ! 1-002 - Error RPG$_INVDATTYP changed to RPG$_INVARG. DG 11-Jul-1983
1 !--
1
1 !<BLF/PAGE>
```

```
%SBTTL 'Declarations'
!+
! PROLOGUE FILE:
!-

REQUIRE 'RTLIN:RPGPROLOG';                              ! Switches, PSECTs, macros,
                                                        !  linkages and LIBRARYs

!+
! LINKAGES
!       NONE
!-

!+
! TABLE OF CONTENTS:
!-

FORWARD ROUTINE
        RPG$SQRT : NOVALUE ;

!+
! INCLUDE FILES
!       NONE
!-

!+
! MACROS
!       NONE
!-

!+
! EQUATED SYMBOLS
!       NONE
!-

!+
! EXTERNAL REFERENCES
!-

EXTERNAL ROUTINE
        COB$CVTID_R7: JSB_67,                           ! Convert CIT to D_floating
        COB$CVTLI_R8: JSB_678,                          ! Convert long to CIT (with scaling)
        COB$CVTPD_R9: JSB_6789,                         ! Convert packed to D_floating
        COB$CVTRDP_R9: JSB_6789,                        ! Convert D_floating to packed
        COB$CVTWI_R8: JSB_678,                          ! Convert word to CIT (with scaling)
        LIB$STOP,                                       ! Stop execution via signalling
        MTH$DSQRT_R5: JSB_D;                            ! Square root of D_floating

EXTERNAL LITERAL
        MTH$_SQUROONEG,                                 ! Square root of negative number
        RPG$_INVARG;                                    ! Invalid data type

EXTERNAL
        LIB$AB_CVTTP_O,                                 ! Table for convert trailing to packed
        RPG$BTZ;                                        ! Table for translate blank to zero
```

```
%SBTTL 'RPG$SQRT - Get square root'
GLOBAL ROUTINE RPG$SQRT(
                        FLAGS,                          ! Translation flag
                        NUMBER: REF BLOCK[,BYTE],       ! Argument for square root operation
                        RESULT: REF BLOCK[,BYTE]        ! Result of square root operation
                        ): NOVALUE=

!++     FUNCTIONAL DESCRIPTION
!
!       This routine supports the RPG SQRT opcode.  It is
!       called once by the compiled code for each occurrence
!       of the SQRT opcode for scalars, or once for each
!       element of an array.
!       It accepts an input number of word, long, packed, or
!       right overpunched numeric data type; and outputs a
!       packed result.
!
! CALLING SEQUENCE:
!
!       CALL RPG$SQRT (flags.rl.v, number.rx.ds, result.wp.ds)
!
! FORMAL PARAMETERS:
!
!       flags           longword integer - bit 1 set if blanks in
!                       overpunched numeric field should be treated
!                       as equivalent to zeroes; otherwise the
!                       translation is not done.
!
!       number          address of descriptor of argument for square
!                       root operation.
!                       The allowable data types are word, long,
!                       packed, and right overpunched numeric.
!
!       result          address of descriptor of result of the square
!                       root operation.
!                       The allowable data type is packed.
!
! IMPLICIT INPUTS:
!
!       NONE
!
! IMPLICIT OUTPUTS:
!
!       NONE
!
! COMPLETION CODES:
!
!       SS$_NORMAL
!
! SIDE EFFECTS:
!
!       If NUMBER is negative, the result field is set to zero and the
!       error MTH$_SQUROONEG is signalled.
!
!--

```

```
;  164          0226  1 !<BLF/PAGE>
```

```

BEGIN

LITERAL
    BTZ_BIT = 2,                                        ! Convert blanks to zeroes
    MAX_PACKED_LEN = 15;                                ! Maximum allowed packed decimal number length

LOCAL
    D_VALUE:            VECTOR[2],                      ! Input number converted to D_floating
    D_SQRT:             VECTOR[2],                      ! D_floating square root result
    I_VALUE:            VECTOR[12, BYTE],               ! COBOL intermediate temporary
    PACKED_LENGTH,
    PACKED_NUMBER:  VECTOR [MAX_PACKED_LEN/2 + 1, BYTE],
                                                        ! Packed decimal number
    SCALE;                                              ! Scale factor

BUILTIN
    CVTTP;                                              ! Convert trailing to packed

!+
!
! Get the scale factor.
!
!-
SCALE = (IF .NUMBER[DSC$B_CLASS] EQL DSC$K_CLASS_SD
            THEN .NUMBER[DSC$B_SCALE]
            ELSE 0);

!+
!
! Convert the input number to D_floating
!
!-
SELECTONE .NUMBER[DSC$B_DTYPE] OF
    SET
    [DSC$K_DTYPE_W]:            ! Word
        BEGIN

        !+
        ! Convert word to CIT to d_floating
        ! (so scale doesn't get lost).
        !-
        COB$CVTWI_R8 (.SCALE, .NUMBER[DSC$A_POINTER], I_VALUE);
        COB$CVTID_R7 (I_VALUE, D_VALUE);

        END;
    [DSC$K_DTYPE_L]:            ! Long
        BEGIN

        !+
        ! Convert long to CIT to d_floating
        ! (so scale doesn't get lost).
        !-
        COB$CVTLI_R8 (.SCALE, .NUMBER[DSC$A_POINTER], I_VALUE);
        COB$CVTID_R7 (I_VALUE, D_VALUE);

        END;
```

```
2               [DSC$K_DTYPE_P]:          ! Packed
2
2                   COB$CVTPD_R9 (.SCALE, .NUMBER[DSC$W_LENGTH], .NUMBER[DSC$A_POINTER], D_VALUE);
2
2               [DSC$K_DTYPE_NRO]:        ! Right overpunched numeric
3                   BEGIN
3
3                   IF (.FLAGS AND BTZ_BIT) NEQ 0
3                   THEN
3                       !+
3                       ! Translate blanks to zeroes if flag set.
3                       !-
3                       CH$TRANSLATE (RPG$BTZ, .NUMBER[DSC$W_LENGTH], .NUMBER[DSC$A_POINTER],
3                                     0, .NUMBER[DSC$W_LENGTH], .NUMBER[DSC$A_POINTER]);
3                   !+
3                   ! Convert trailing to packed to d_floating.
3                   !-
3                   PACKED_LENGTH = MAX_PACKED_LEN;
3                   CVTTP (.NUMBER[DSC$W_LENGTH], .NUMBER[DSC$A_POINTER], LIB$AB_CVTTP_O, PACKED_LENGTH, PACKED_NUMBE
3                   COB$CVTPD_R9 (.SCALE, MAX_PACKED_LEN, PACKED_NUMBER, D_VALUE);
3
3                   END;
2               [OTHERWISE]:
2
2                   LIB$STOP (RPG$_INVARG);
2
2               TES;
2
2           !+
2           !
2           ! Take the square root of the D_floating value and
2           ! convert the result to the output data type (packed)
2           !
2           !-
2           MTH$DSQRT_R5 (.D_VALUE[0], .D_VALUE[1]; D_SQRT[0], D_SQRT[1]);
3           SCALE = (IF .RESULT[DSC$B_CLASS] EQL DSC$K_CLASS_SD
3                   THEN .RESULT[DSC$B_SCALE]
2                   ELSE 0);
2           COB$CVTRDP_R9 (-.SCALE, D_SQRT, .RESULT[DSC$W_LENGTH], .RESULT[DSC$A_POINTER]);
2
1           END;


                                                .TITLE  RPG$SQRT Get square root
                                                .IDENT  \1-002\

                                                .EXTRN  COB$CVTID_R7, COB$CVTLI_R8
                                                .EXTRN  COB$CVTPD_R9, COB$CVTRDP_R9
                                                .EXTRN  COB$CVTWI_R8, LIB$STOP
                                                .EXTRN  MTH$DSQRT_R5, MTH$_SQUROONEG
                                                .EXTRN  RPG$_INVARG, LIB$AB_CVTTP_O
                                                .EXTRN  RPG$BTZ

                                                .PSECT  _RPG$CODE,NOWRT,  SHR,  PIC,2

                          0FFC 00000            .ENTRY  RPG$SQRT, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-; 0170
                                                        R11                                         ;
```

```
                                5E          24  C2 00002      SUBL2   #36, SP                                     ;
                                5A    08    AC  D0 00005      MOVL    NUMBER, R10                                 ; 0251
                                09    03    AA  91 00009      CMPB    3(R10), #9                                  ;
                                            06  12 0000D      BNEQ    1$                                          ;
                                5B    08    AA  98 0000F      CVTBL   8(R10), SCALE                               ; 0252
                                            02  11 00013      BRB     2$                                          ;
                                            5B  D4 00015 1$:  CLRL    SCALE                                       ; 0251
                                50    02    AA  9A 00017 2$:  MOVZBL  2(R10), R0                                  ; 0260
                                07          50  91 0001B      CMPB    R0, #7                                      ; 0262
                                            13  12 0001E      BNEQ    3$                                          ;
                                58    08    AE  9E 00020      MOVAB   I_VALUE, R8                                 ; 0269
                                57    04    AA  D0 00024      MOVL    4(R10), R7                                  ;
                                56          5B  D0 00028      MOVL    SCALE, R6                                   ;
                           00000000G        00  16 0002B      JSB     COB$CVTWI_R8                                ;
                                            16  11 00031      BRB     4$                                          ; 0270
                                08          50  91 00033 3$:  CMPB    R0, #8                                      ; 0273
                                            21  12 00036      BNEQ    5$                                          ;
                                58    08    AE  9E 00038      MOVAB   I_VALUE, R8                                 ; 0280
                                57    04    AA  D0 0003C      MOVL    4(R10), R7                                  ;
                                56          5B  D0 00040      MOVL    SCALE, R6                                   ;
                           00000000G        00  16 00043      JSB     COB$CVTLI_R8                                ;
                                57    1C    AE  9E 00049 4$:  MOVAB   D_VALUE, R7                                 ; 0281
                                56    08    AE  9E 0004D      MOVAB   I_VALUE, R6                                 ;
                           00000000G        00  16 00051      JSB     COB$CVTID_R7                                ;
                                            58  11 00057      BRB     10$                                         ; 0260
                                15          50  91 00059 5$:  CMPB    R0, #21                                     ; 0284
                                            0D  12 0005C      BNEQ    6$                                          ;
                                59    1C    AE  9E 0005E      MOVAB   D_VALUE, R9                                 ; 0286
                                58    04    AA  D0 00062      MOVL    4(R10), R8                                  ;
                                57          6A  3C 00066      MOVZWL  (R10), R7                                   ;
                                            2E  11 00069      BRB     8$                                          ;
                                13          50  91 0006B 6$:  CMPB    R0, #19                                     ; 0288
                                            34  12 0006E      BNEQ    9$                                          ;
                      0D        6C          21  E1 00070      BBC     #33, FLAGS, 7$                              ; 0291
00000000G  00         00  04    BA          6A  2E 00074      MOVTC   (R10), @4(R10), #0, RPG$BTZ, (R10), @4(R10) ; 0297
                          04    BA          6A     0007E                                                          ;
                                50          0F  D0 00081 7$:  MOVL    #15, PACKED_LENGTH                          ; 0301
       50 00000000G  00   04    BA          6A  26 00084      CVTTP   (R10), @4(R10), LIB$AB_CVTTP_O, -           ; 0302
                                            6E     0008E              PACKED_LENGTH, PACKED_NUMBER                ;
                                59    1C    AE  9E 0008F      MOVAB   D_VALUE, R9                                 ; 0303
                                58          6E  9E 00093      MOVAB   PACKED_NUMBER, R8                           ;
                                57          0F  D0 00096      MOVL    #15, R7                                     ;
                                56          5B  D0 00099 8$:  MOVL    SCALE, R6                                   ;
                           00000000G        00  16 0009C      JSB     COB$CVTPD_R9                                ;
                                            0D  11 000A2      BRB     10$                                         ; 0260
                           00000000G        8F  DD 000A4 9$:  PUSHL   #RPG$_INVARG                                ; 0308
                 00000000G  00              01  FB 000AA      CALLS   #1, LIB$STOP                                ;
                                50    1C    AE  7D 000B1 10$: MOVQ    D_VALUE, R0                                 ; 0318
                           00000000G        00  16 000B5      JSB     MTH$DSQRT_R5                                ;
                           14   AE          50  7D 000BB      MOVQ    R0, D_SQRT                                  ;
                                50    0C    AC  D0 000BF      MOVL    RESULT, R0                                  ; 0319
                                09    03    A0  91 000C3      CMPB    3(R0), #9                                   ;
                                            06  12 000C7      BNEQ    11$                                         ;
                                5B    08    A0  98 000C9      CVTBL   8(R0), SCALE                                ; 0320
                                            02  11 000CD      BRB     12$                                         ;
                                            5B  D4 000CF 11$: CLRL    SCALE                                       ; 0319
                                57    14    AE  9E 000D1 12$: MOVAB   D_SQRT, R7                                  ; 0322
```

```
                                        56          5B  CE 000D5          MNEGL    SCALE, R6                        ;
                                        59      04  A0  D0 000D8          MOVL     4(R0), R9                        ;
                                        58          60  3C 000DC          MOVZWL   (R0), R8                         ;
                                          00000000G 00  16 000DF          JSB      COB$CVTRDP_R9                    ;
                                                        04 000E5          RET                                       ; 0324

; Routine Size:  230 bytes,     Routine Base:  _RPG$CODE + 0000


;  264          0325  1
;  265          0326  0 END ELUDOM




;                                       PSECT SUMMARY
;
;        Name                       Bytes                       Attributes
;
;  _RPG$CODE                             230  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)


;                                Library Statistics
;
;                                               -------- Symbols --------        Pages        Processing
;        File                                   Total    Loaded   Percent        Mapped       Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1             9776        10         0          581          00:00.9
;  _$255$DUA28:[RPGRTL.OBJ]RPGLIB.L32;1            54         4         7            9          00:00.1




;                                       COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:RPGSQRT/OBJ=OBJ$:RPGSQRT MSRC$:RPGSQRT/UPDATE=(ENH$:RPGSQRT)

; Size:             230 code + 0 data bytes
; Run Time:            00:06.1
; Elapsed Time:        00:18.3
; Lines/CPU Min:         3217
; Lexemes/CPU-Min: 13430
; Memory Used:   91 pages
; Compilation Complete
```

0332 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY